Imprint-Based Input Techniques for Touch-Based Mobile Devices

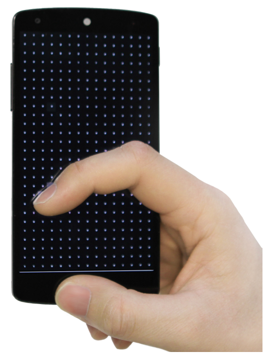
Huy Viet Le
mail@huyle.de
University of Stuttgart
Stuttgart, Germany

Sven Mayer
info@sven-mayer.com
LMU Munich
Munich, Germany

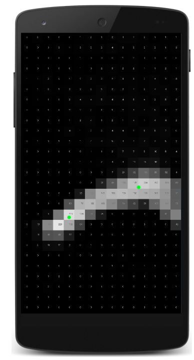
Niels Henze
niels.henze@ur.de
University of Regensburg
Regensburg, Germany

ABSTRACT

Touchscreens translate touches of all kinds into 2D coordinates. This limits the input vocabulary and constrains effective interaction to touches by the fingertip. Previous tabletop research extended the input vocabulary with a myriad of promising input techniques using the shape of fingers and hands. However, these techniques are not applicable to mobile devices due to differences in size, ergonomics, and technology. We conducted ideation sessions (N=17) to explore novel input techniques and use cases for *imprint-based touch sensing* on mobile devices. As a case study, we present *FlexionTouch*, a novel input technique that recognizes the finger flexion on a touchscreen. Using the finger flexion as an additional input dimension, *FlexionTouch* provides an always-available shortcut and can be used for value inputs, document previews, and gestures. We propose five example use cases for *FlexionTouch* input which we evaluated in a second user study (N=20). While the low resolution of the capacitive images leads to a less accurate input compared to tabletops, participants still find the presented use cases helpful. As our input technique is purely software-based, it can be readily deployed to every mobile device with a capacitive touchscreen.



(a) Finger placed on the display



(b) Raw Data

Figure 1: Extending touch input using the shape of touches in addition to 2D coordinates that commodity touchscreens provide (see lime points). These images showcase FlexionTouch, an additional input method that maps the flexion of a finger placed flatly on the display to a continuous value (e.g. low flexion correspond to a low value and vice versa).

CCS CONCEPTS

• **Human-centered computing** → **Touch screens**; Empirical studies in HCI; • **Hardware** → *Touch screens.*

KEYWORDS

touchscreen, finger, flexion, input

ACM Reference Format:

Huy Viet Le, Sven Mayer, and Niels Henze. 2020. Imprint-Based Input Techniques for Touch-Based Mobile Devices. In *19th International Conference on Mobile and Ubiquitous Multimedia (MUM 2020), November 22–25, 2020, Essen, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3428361.3428393

1 INTRODUCTION AND RELATED WORK

Touchscreens enable intuitive interactions by combining input and output in a single interface. By translating touches into 2D coordinates, users can directly touch elements of the user interface (UI) and interact with them similar to physical objects. Despite these advantages, touch input is still inferior to traditional input devices such as mouse and keyboard due to the limited input vocabulary. While a traditional mouse provides two or more buttons to activate different functions at the same 2D coordinate, the expressiveness of a touch is limited to simple 2D coordinates. This limitation slows down the interaction and with less options to provide shortcuts. Today's touch input even contradicts Shneiderman's golden rules [53].

To extend the touch input vocabulary, previous work presented a wide range of approaches predominantly based on either 2D gestures [3, 25, 33, 35, 46, 47, 51, 52] additional sensors that need to be integrated into the device [18, 20–22, 34, 41, 50, 58], or worn by the user [13, 14, 36, 37, 55]. These extensions of touch input are either limited or greatly reduce mobility and convenience since additional sensors need to be integrated into the mobile device or worn by the user. In contrast, research in the field of tabletops extensively use images of touches (*i.e.*, an image of the imprint of the finger or hand on a touch-sensitive surface) to extend the input vocabulary by using the whole contact area of fingers [2, 9, 19], hand [5, 8, 38], and even beyond (*e.g.*, forearm [26], arm location [1]). For example, the hand's contact shape was used for controlling menus and continuous values [38], while the hand posture was used to augment touch input [19]. The concept of using the contact surface of the finger imprint extends the input vocabulary in a natural way since the whole finger and hand can be used to manipulate virtual objects.

MUM 2020, November 22–25, 2020, Essen, Germany

ACM ISBN 978-1-4503-8870-2/20/11…$15.00
https://doi.org/10.1145/3428361.3428393

In this work, we will refer to this concept as *imprint-based touch sensing*.

Input techniques that consider the contact surface require sensors which provide virtual imprints of touches. While tabletops and other stationary devices are large enough to include cameras (*e.g.*, RGB [56, 61], infrared [16, 43, 57], depth [27, 40], and the Leap Motion [6]) below or around the touch surface, these sensors are impractical for mobile devices such as smartphones. Recently, researchers modified the firmware of commodity smartphones to access the raw data of the mutual capacitive touchscreen which represent low-resolution finger imprints. Previous work referred to them as *capacitive images* [12, 23, 28, 29, 32, 39] and showed that they contain sufficient signal to identify body parts [23], palm touches [28], hand poses [42], finger knuckle [54], finger orientations [39, 59], and even different fingers [10, 32]. While these contributions infer additional properties of touches based on machine learning, none of them used the actual size and shape of contact surfaces to explore novel interaction methods for mobile devices similar to the ones for tabletops. Most related are previous work by Oakley *et al.* [45] who presented imprint-based gestures on smartwatches, and Boring *et al.* [4] who used the contact size for panning and zooming.

To benefit from imprint-based input methods presented for tabletops [7], we need to port them to mobile devices such as smartphones and tablets. However, simply applying these input methods to smartphones is not enough. Not only are smartphones smaller and provide lower resolution of touches, users also prefer to use the same hand for holding and interacting [24, 44] which leads to limited finger movements. Moreover, common use cases for smartphones differ from the ones for tabletops (*e.g.*, different use cases and applications; thus, different challenges to solve) which also requires an additional exploration of use cases and scenarios. Thus, it is important to understand the users needs and wants to design and adapt new input methods.

In this paper, we present the results of three ideation sessions (N=17) about novel input techniques based on *imprint-based touch sensing* on mobile devices as a first step. In particular, we focus on input techniques which potential users envision and their benefit in common mobile use cases and scenarios. One outcome of the ideation sessions are use cases and concepts of novel imprint-based input techniques. As a second step, we implemented one of the proposed input technique as a case study which we refer to as *FlexionTouch*. Our input technique determines the flexion of a finger placed on the display based on the touch imprint and uses the flexion as an additional input dimension. This enables a wide range of use cases including the use as a shortcut, an always-available slider (*e.g.*, for screen brightness and device volume), and as a preview method for various media. The contribution of this paper is thus two-fold: (1) results of ideation sessions on novel imprint-based input techniques for mobile devices using the contact shape and area; and (2) the technical details, implementation, and evaluation of *FlexionTouch*.

2 STUDY 1: IDEATION SESSIONS

Previous work on imprint-based input techniques predominantly focused on tabletops. However, they never entered the mobile device domain. To close this gap we conducted ideation sessions following the practitioner's guide by Gvero *et al.* [15]. The goal was to explore the advantages and disadvantages of imprint-based input techniques on mobile devices. Additionally, we wanted to explore possible use cases for imprint-based input.

2.1 Participants

To involve technically proficient users with knowledge about current input techniques on mobile devices, we recruited 17 participants (13 male and 4 female) between the ages of 20 and 30 (M = 24.9, SD = 3.1) through mailing lists and word of mouth. Participants work and/or study at a university in central Europe (12) or North America (5) and have a technical or design background. All participants are reportedly users of recent smartphones and consider themselves as experienced users. Participants were split into three sessions (with 7, 5, and 5 participants) which took part separately. We reimbursed them with 10 EUR for their participation.

2.2 Procedure

After obtaining informed consent and gathering demographic information, we briefed participants on the process of the ideation session and introduced them to the background of mobile touch interaction. The ideation session comprises four questions which were introduced and explained by the moderator. The questions were ordered in a top-down scheme; starting with abstract topics and concluding with concrete questions. All questions were answered by individual participants first (3 min), then discussed in pairs (3 min), and eventually discussed within the whole group under the lead of the moderator (4 min). Participants noted their answers onto sticky notes, attached them to a board, and clustered them after each question.

As the first question, we asked participants to (**Q1**) *think about input methods beyond touch input with 2D coordinates.* The moderator explained the limitation of recent touch input to 2D coordinates and the implications (*i.e.* limited input vocabulary, lack of shortcuts, and reachability problems). After collecting a broad range of input methods beyond 2D touch, we asked participants to (**Q2**) *think about input methods that use the contact shape and size of whole fingers.* To give them a concrete vision on the use of a whole finger, we showcased the capacitive images captured on an LG Nexus 5 and passed the device around. Afterward, we asked them about (**Q3**) *potential use cases for the input methods that were collected previously.* Finally, we discussed with participants (**Q4**) *how such input methods would help them in daily use and whether they would personally use it.* Each session took 30 minutes.

2.3 Ideation Session Results

We transcribed the hand-written answers, separated multiple answers written onto a single sticky notes, and printed them as paper cards. Three researchers then employed a simplified version of qualitative coding with affinity diagramming [17] by discussing and clustering all answers thematically. We present the thematic clusters in the following.

2.3.1 Q1: Input Methods beyond 2D Coordinates. Beyond the concept of translating touches to 2D coordinates (including dwell times and gestures), participants envisioned the recognition of hand grips to augment touch input. This could be done by extending touch sensing beyond the touchscreen (*i.e.* to the device surface [29, 31]) and even with the built-in accelerometer [11]. Moreover, participants thought about differentiating inputs by fingertips and the whole finger (*e.g.*, scrolling vertically with the whole finger for switching between apps). Besides touch input, participants also suggested the use of external sensors such as voice input, physiological sensing (*e.g.* heartbeat sensor), additional cameras for mid-air gestures, as well as external mice and keyboards.

2.3.2 Q2: Input Methods based on Contact Shape and Size. We asked participants about input methods that use the finger's contact shape and size. All participants mentioned swipe gestures using a flat finger (*i.e.* placing the full finger horizontally on the display and swiping up/down) as the most obvious input method. Extending this idea, they also suggested to swipe with two flat fingers (*e.g.* thumb of both hands) and using the distance between them to enter a continuous value. Beyond swiping, they also envisioned finger rolling; means placing the full finger on the display and rotating it along the longitudinal axis. This is similar to previous work by Roudaut *et al.* [52] who did that for the thumb tip. Since the size of the contact area could act as a proxy for the finger's pitch angle (*i.e.* the angle between finger and horizontal touch surface) when touching the display with the fingertip, participants suggested a binary input modality which differentiates between finger touches perpendicular and nearly parallel to the display. Similarly, the contact area could also represent the pressure of a touch based on the deformation of the fingertip skin (similar to *The Fat Thumb* [4]).

While initially suggested by the minority of participants, the idea of using the finger flexion ("bentness" as mentioned by participants) quickly became one of the most discussed ideas within the group discussion of all sessions. Thereby, participants quickly realized that the flexion angle could be used to enter binary, nominal, as well as continuous values (0 for stretched finger, 100 for fully flexed finger).

2.3.3 Q3: Use Cases. While participants collected use cases for all input methods gathered in the previous part, we focus on full finger swipes and the finger flexion as these are the most discussed and promising input methods. Swiping up/down with the whole finger was envisioned as a shortcut to switch between applications while swiping to the right could close them (*c.f.* swiping the application away). Similarly, swipe up/down movement was envisioned as a metaphorical gesture to switch between layers in the application (*c.f.* peeling off the layers). Moreover, they discussed the full finger as a way to select text by wiping over the desired lines.

A use case for text selection was also proposed for the finger flexion. Instead of selecting both starting and end point of the selection (leading to occlusion during selection), the finger flexion could be used to change the selection length (starting from an initial point) to avoid occlusion using indirect input. Participants also suggested continuous changes of the finger flexion to enter values. However, instead of replacing direct touch (which was reportedly perceived as the most intuitive and fast method) for value input, finger flexion was envisioned as a shortcut to enter values. Remembering certain finger flexion levels and directly placing them on the display could be used to enter values instantly instead of changing slider, knobs, or entering values using a virtual keyboard. Moreover, this represents an always-available method to enter values for functions such as changing the device volume or screen brightness. Participants also thought about using finger flexion for games; such as directly chopping a fruit in *Fruit Ninja* instead of performing a swipe gesture.

2.3.4 Q4: Benefits of elicited input methods. When asked for benefits in daily use, participants were unanimous about the use as a shortcut. A wide range of frequently used functions was mentioned that are currently inconvenient to access since multiple touches are required (*e.g.*, application switching, changing display brightness, launching applications). However, participants were also unanimous about direct touch being the most intuitive and easiest way to enter values. Thus, participants agreed that the proposed input methods should be an addition to direct touch but not replace it.

2.4 Discussion

The results of the ideation sessions include a number of promising input methods that extend the currently limited touch input vocabulary: (1) performing swipe gestures with a flat finger, (2) recognizing finger rolling, (3) using the size of the fingertip contact area, and (4) mapping the flexion of a finger to a discrete value. Differentiating between touches of a fingertip and a flat finger (*e.g.*, to determine the finger flexion) could already be used as a shortcut (e.g., using an area threshold or machine learning). While participants were unanimous about the input method's usefulness as a shortcut, they argued that it should be used as an extension instead of a direct touch replacement. This is due to the superior intuitiveness of direct touch.

The four input methods described above (as well as a number of further suggestions) could be readily implemented even with the low-resolution touch images of smartphones. While input methods based on the concept of *imprint-based touch sensing* could extend the input vocabulary without interfering with common touch input that uses the fingertip, there is no previous work that investigated such input methods on smartphones. This is surprising as capacitive images are available on nearly all mass-market smartphones with a mutual capacitive touchscreen so that such input methods could be readily deployed via software updates.

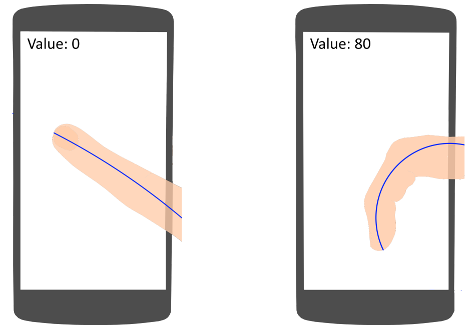
That capacitive images are currently not widely used for novel input techniques could be due to two reasons: First, capacitive images are not commonly exposed to the application layer so that a modified system kernel is currently required. However, SDKs of common mobile operating systems (*e.g.*, Android, iOS) already provide the radius and the estimated pressure of a touch which is a first step towards using area and shape of touches. Moreover, such a modified kernel could be easily distributed via over-the-air updates. Second, capacitive touchscreens only provide low-resolution images of touches (*e.g.*, 15×27 on an LG Nexus 5) which makes it difficult to implement precise interactions. However, touch controllers show that touches can be condensed to precise and jitterless touch coordinates. Moreover, devices based on Microsoft's PixelSense technology provide high-resolution touch images (*e.g.*, $540 \times 960\,px$ on the Microsoft PixelSense 2) based on IR sensing integrated into the LCD layer. As this technology does not require additional cameras, it could be used on mobile devices. Tablet-PCs such as the Microsoft Surface Pro 4 already have the PixelSense technology.

With *imprint-based touch sensing* becoming increasingly accessible on mobile devices, we believe that input methods based on full hands and fingers provide exciting opportunities to improve the interaction with future mobile devices. Unfortunately, research results from previous tabletop research cannot be simply applied to mobile devices. Compared to stationary touchscreen devices such as tabletops, smartphones and tablets have a smaller display. Additional challenges emerge since users commonly use the same hand for holding and interacting with the device [24, 44] which limits feasible finger movements [30]. Moreover, the use cases for such input methods might differ between tabletops and mobile devices since different applications are used (e.g., more applications that are used while on the move). Thus, an important basis to inform the design such input methods is to understand how users want to use them, for which use cases, and how they perform the finger movements.

3 SHOW CASE: FLEXIONTOUCH – A NOVEL INPUT METHOD

To show that *capacitive images* can indeed be used for imprint-based input techniques that rely on the size and shape of contact surfaces, we developed an input method as a case study. For the case study, we implemented an input method based on the finger flexion which was designed in the ideation sessions. We refer to this input method as *FlexionTouch* which extends touch input by using the flexion of a finger placed on the display to enter continuous values. Touches of a full finger can be differentiated from fingertip touches by either using simple area thresholds or machine learning approaches similar to previous work [28, 54]. In the following, we present the concept, technical details, and implementation.

FlexionTouch maps the degree of finger flexion to continuous integer values; examples are shown in Figure 2. A low value can be produced with a stretched finger on the display while a flexed finger produces a high value. In all our examples, the lowest value is 0 and the highest is 100. Changing the finger flexion while in contact with the display changes the value continuously.



(a) Low Flexion **(b) High Flexion**

Figure 2: A straight finger (low flexion) produces a low value, and a flexed (high flexion) finger is mapped to a high value.

3.1 Use Cases

Bases on our initial ideation sessions, we present use cases which are tailored to fit our show case of using *FlexionTouch* as an additional imprint-based input for mobile devices.

3.1.1 Entering Values with a Single Touch. In contrast to established mechanisms for entering continuous values (*e.g.*, sliders, keyboard, etc.) that require multiple gestures/taps (*e.g.*, dragging to desired value, typing multiple digits), placing a finger in a certain angle on the display immediately results in a value. We envision that experienced users can memorize the finger flexion for particular values in their muscle memory and thus enter values with a single touch.

3.1.2 Always-Available Value Input. Frequently used functions (*e.g.*, changing screen brightness) requires opening a menu. *FlexionTouch* enables users to enter a value within any application without the need of a dedicated UI element. The screen brightness could be adjusted in any state with the finger flexion.

3.1.3 Flexion Gestures. Since a fingertip produces a noticeably smaller contact area than a full finger, the recognition thereof could also be used as a binary gesture, similar to previous work that identified different input sources [23, 28]. Moreover, changes in flexion values could also be used to provide one-dimensional gestures. For example, a finger-stretching gesture could be mapped to closing an application (metaphorically flicking the app away) while a finger-flexion gesture could be mapped to save a file (metaphorically wiping the file closer to the user).

3.1.4 Document Preview. The continuous change of the input value based on the change of the flexion angle could also be used to skim through documents. For example, the range from stretched to flexed could be mapped to all pages of a document. Thereby, users can then slowly stretch the finger to get a fast preview of the document. This concept could also be applied to other media, such as videos, image galleries, pre-listen to songs, or games.

3.1.5 Porous Interfaces. Gupta *et al.* [13] presented *Porous User Interfaces*, which is a concept that overlays two applications. Thereby, one finger interacts with the application in the foreground while the other interacts with the background application. Similarly, touches with the fingertip can be used to interact with the "main layer" of an application while *FlexionTouch* could be used to set values in a "secondary layer". With this, both layers can be overlapped visually while users can interact with both.

3.2 Mobile Implementation

We used an LG Nexus 5 with a modified kernel to access the 15×27 8-bit raw capacitive image of the Synaptics ClearPad 3350 touch sensor at 20 *fps* as described in previous work [29, 31, 59]. An example of the raw data is shown in Figure 1b, where each image pixel corresponds to a $4.1\,mm \times 4.1\,mm$ square on the $4.95''$ touchscreen. The pixel values represent the differences in electrical capacitance (in pF) between the baseline measurement and the current measurement. The processing pipeline to translate a capacitive image to a continuous value is shown in Figure 3. All processing steps were implemented in C based on OpenCV and compiled with the Android NDK to reduce the runtime. Converting a capacitive image to a continuous value takes 1.16 *ms* on average (*SD* = 1.08 *ms*, *min* = 0.08 *ms*, *max* = 8.65 *ms*).

(1) *Noise Removal and Upscaling*: We removed noise in the capacitive image (due to electromagnetic interference with, *e.g.*, the LCD) using a 2D filter with a 5×5 kernel. We then upscaled the capacitive image with a factor of 7 which we empirically determined to provide the best trade-off between computational time and accuracy, see Figure 3b.
(2) *Image Thinning*: We then used the Zhang-Suen thinning algorithm [60] on a thresholded version of the capacitive image to convert it to a skeleton representing the finger curvature, see Figures 3c and 3d.
(3) *Circle Fitting and Value Mapping*: Based on the thinned capacitive image, we fitted a circle using the Pratt method [49] which stays robust even if data points are observed only within a small arc. We then use the radius of the circle to describe the finger flexion, mapped to a range between 0 and 100. While straight fingers produce larger radii, they also lead to noticeable jitter for little movements. Thus, we used a maximum radius threshold to avoid jitter. While we experimented with different mapping functions, we found that a linear mapping function worked best. We experimented with different conversion approaches (*e.g.*, fitting a B-spline and calculating the average angles between their anchors, fitting quadratic curves with a rotation parameter) and empirically found that the circle fitting approach worked best with the lowest jitter, see Figure 3e.

4 STUDY 2: EVALUATING FLEXIONTOUCH

We conducted a user study to evaluate *FlexionTouch*. While we compare *FlexionTouch* with direct touch to collect quantitative measures, we focused on the qualitative feedback by potential users on the perceived usefulness of *FlexionTouch*.

4.1 Participants

We recruited 20 participants (18 male and 2 female) between the ages of 20 and 31 (M = 23.6, SD = 3.0). All participants were right-handed. The average hand size was measured from the wrist crease to the middle fingertip and ranged from 16.0 *cm* to 25.0 *cm* (M = 19.6 *cm*, SD = 2.1 *cm*). Our collected data comprise samples from the 5th and 95th percentile of the anthropometric data reported in prior work [48]. Thus, the sample can be considered as representative.

4.2 Study Procedure and Design

After obtaining informed consent, we briefed participants about the input method, gathered demographic information, and handed them an instruction sheet which explained all tasks. Participants were then instructed to practice *FlexionTouch* for entering continuous values on a slider, see Figure 4a. After the initial practice in which we ensured that participants were familiar with *FlexionTouch*, we demonstrated the technique and asked them to test multiple use cases; including performing flexion and stretch gestures to copy and paste text (see Figure 4b), performing flexion-gestures (see Figure 4c), and entering values in one step by placing the already flexed finger on the display. When everything was understood by the participants, they solved 1D target selection tasks in which they set the horizontal position of the slider knob to the displayed target. Participants saw continuous feedback as the slider knob moved with their input. Finally, we interviewed participants on their impressions about *FlexionTouch*, advantages and disadvantages, further potential use cases, and whether they would use *FlexionTouch* on their own device.

We used a 2×2 within-subjects design for the 1D target selection task with HANDS (one-handed vs. two-handed) and INPUT METHOD (Direct Touch vs. *FlexionTouch*) as independent variables. Each task consists of a seek bar at a random vertical position for which participants are instructed to move the knob as precise and as fast as possible to the red target. Thereby, we compared values entered using *FlexionTouch* with a direct touch slider as commonly implemented on mobile user interfaces. Our dependent variables are the task completion time (begin: task shown; end: correct value entered) and the accuracy (i.e. value offsets). We randomized the order of the four conditions using a Latin square to avoid sequence effects. The study took around 30 minutes with optional breaks. Additionally, participants were free in how they use and hold the phone.

4.3 Apparatus

We used the LG Nexus 5 and the implementation as described above. We integrated *FlexionTouch* into a custom application for the study tasks, see Figure 4. Interviews were recorded with a handy recorder.

5 RESULTS

We present the quantitative and qualitative feedback based on our evaluation study in which we asked participants to use *FlexionTouch* – a show case of imprint-based input techniques to enrich input for mobile devices.

5.1 Task Completion Time

We conducted a two-way repeated measures analysis of variance (RM-ANOVA) to determine whether HANDS and INPUT METHOD significantly influenced the task completion time (TCT). Our analysis revealed statistically significant main effects for HANDS on TCT ($F_{1,19}$ = 12.890, p = .021). However, no statistically significant main effects for INPUT METHOD ($F_{1,19}$ = 0.332, p = .571) and no two-way interaction effect between HANDS and INPUT METHOD ($F_{1,19}$ = 0.239, p = .630). On average, participants needed 3.0 *s* (*SD* = 1.9 *s*) with Direct Touch and 3.1 *s* (*SD* = 2.3 *s*) using *FlexionTouch*. For the one-handed condition, Direct Touch took 3.2 *s* (*SD* = 2.2 *s*) and *FlexionTouch* took 3.4 *s* (*SD* = 2.7 *s*) while for the two-handed condition, Direct Touch took 2.8 *s* (*SD* = 1.6 *s*) and *FlexionTouch* took 2.8 *s* (*SD* = 1.8 *s*) on average.

5.2 Accuracy

We conducted a two-way RM-ANOVA to determine whether HANDS and INPUT METHOD significantly influenced the input accuracy. Our analysis revealed statistically significant main effects for HANDS on *accuracy* ($F_{1,19}$ = 6.398, p < .002) and a statistically significant main effects for INPUT METHOD ($F_{1,19}$ = 220.159, p < .001). Our

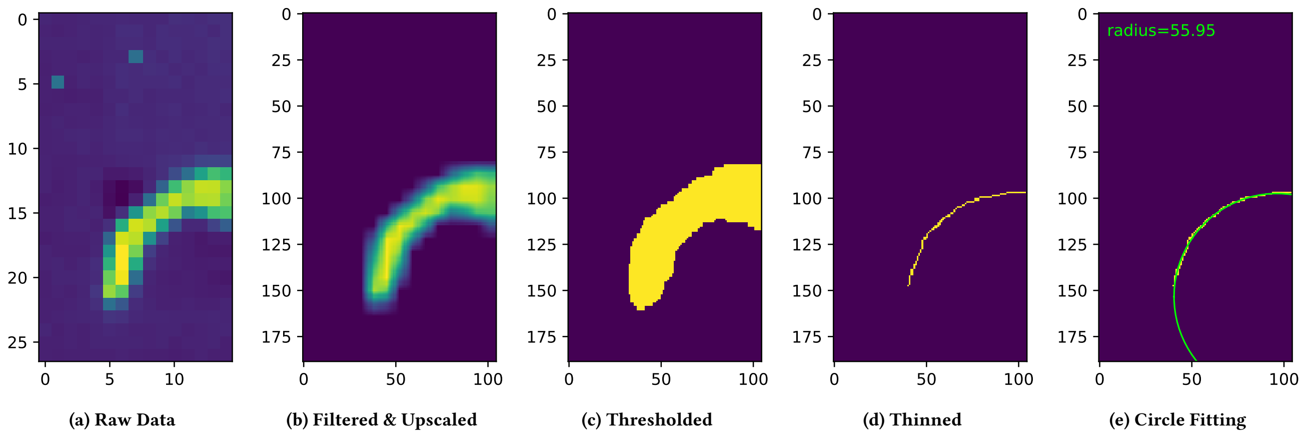


Figure 3: Pipeline for translating the capacitive image of a flatly placed finger to a value which describes the flexion of the finger. Figure (a) shows the original capacitive image, Figure (b) shows the filtered and 7× upscaled version of of the capacitive image, Figure (c) shows a thresholded version which is then thinned in Figure (d). Figure (e) fits a circle using the Pratt method [49] to map the circle radius to a flexion value as described in Figure 2.

analysis revealed a two-way interaction effect between HANDS and INPUT METHOD ($F_{1,19} = 4.959$, $p = .039$). On a slider with values between 0 and 100, on average, the error was 1.3 ($SD = 3.5$) in the Direct Touch condition and 13.0 ($SD = 15.8$) using *FlexionTouch*.

5.3 Qualitative Feedback

Two researchers employed a simplified version of qualitative coding with affinity diagramming [28] on the transcribed interviews by coding the answers, printing them on paper cards, and finally clustering the answers.

5.3.1 First Impression. We asked participants for their first impressions after using *FlexionTouch* during the study. In general, participants perceived *FlexionTouch* as a helpful (*e.g.*, "*I found that it could be very helpful*" - P5) and convenient (P2, P5, P9, P12, P17) input method especially for shortcuts. However, they also emphasized that it should not replace direct touch but rather used as an additional input method for shortcuts ("*If I want to move something on the display, then I think that it is easier to directly touch it, but if I want to use an additional function then [the] input method is really convenient*" - P2). As expected, using the whole finger for input is unfamiliar so that participants expect more time to get familiar and faster with it ("*At the beginning, it was unfamiliar and difficult, but with some practice it gets much better*" - P10; "*I really liked the idea but I found it frustrating in the beginning until I managed to do what I wanted.*" - P18). This is especially the case for the one-handed use of *FlexionTouch* ("*With one hand, it is more difficult than with two hands*" - P9); however, some participants also had the opposite opinion that it gets more inconvenient with two hands since both are required ("*It is a bit unhandy with two hands, but with one hand it is okay*" - P14). Three participants found that it is generally difficult in terms of ergonomics even with practice (P2, P3, P4).

5.3.2 Perceived Advantages and Disadvantages. In total, we identified 15 comments representing advantages and 14 comments indicating disadvantages. The perceived advantages can be clustered into three categories: the usefulness for shortcuts (9), being faster than direct touch (4), and better reachability with *FlexionTouch* than Direct Touch when the seek bar is close to the top edge (3). Participants appreciated the use of *FlexionTouch* as a shortcut for copying and pasting ("*It offers an additional possibility to activate functions without needing to press buttons" - P2*) and for functions that are difficult to reach in nested menus ("*the input method could be faster when used to activate functions that are only reachable within submenus otherwise*" - P7). Four comments mentioned that *FlexionTouch* was perceived as fast although not as accurate ("*It was not as accurate [as direct touch] but as a trade-off really fast to enter values.*" - P8). Moreover, participants found that *FlexionTouch* could help to improve the reachability ("*if I use the phone with one hand, I use my thumb which is not so long to reach the upper left corner on large phones. In this case, the new input method is better than the normal one*" - P5).

The perceived disadvantages include the input accuracy (7). Thus, participants see benefits of *FlexionTouch* only for lowest and highest values (3). Further, a second hand is needed due to the grip instability (4). The accuracy is mostly the result of the high sensibility to movements ("*input was difficult since the value jumped even at slight changes of the flexion*" - P12) and jitter when trying to be precise ("*the more I want to be precise, the more difficult it gets*" - P2) which are both results of the low resolution of capacitive images. Thus, these participants argue that *FlexionTouch* is only beneficial for the lowest and highest value since a finger could be placed fully stretched and bent without re-adjustments ("*I think the input method is only beneficial for the edge values*" - P13). Besides the technical limitations, some participants could not find a stable grip when using *FlexionTouch* one-handedly ("*it is nearly impossible to*

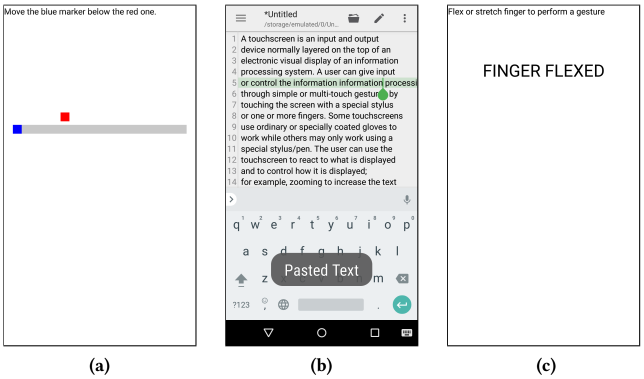


Figure 4: Screenshots of the study application: (a) 1D target selection task for both FlexionTouch and Direct Touch, (b) demo for pasting text using a finger-flexion gesture, and (c) abstract demo for practising finger-flexion gestures.

set the values one-handedly since I have to focus on not dropping the smartphone" - P3).

When asked what they would improve, participants suggested to reduce the amount of values to reduce jitter (4 comments), improve the input accuracy (7 comments), and extend with pre-touch sensing ("*[...] that my finger should also be detected above the display so that I do not have to fully place the finger*" - P4). In contrast, two participants (P2, P13) simply argued that with more practice time, they would be able to achieve a better performance.

5.3.3 How and whether participants would use FlexionTouch. We identified six clusters indicating the use case or scenario for which participants would use *FlexionTouch*. The largest clusters comprise the use as shortcuts (13 comments), whereas six envision general shortcuts (*e.g.*, "*using it for shortcuts of which there are a lot on computers but we do not know on smartphones*" - P3; "*system-wide that are always available*" - P19; "*could be something like a right-click on computers*" - P14), and seven on shortcuts for text editing ("*I like the paste example and could imagine more of them for text copying*" - P11). Moreover, four comments suggested using *FlexionTouch* for scrolling through a text document or video ("*for scrolling or to fast-forward when watching a movie*" - P16). Similarly, four comments suggested to use the input method for previewing documents or videos ("*to skim through documents with the input method instead of pressing a button*" - P15). Three comments envisioned the input method to improve text selection ("*to select text like.. you're somewhere in the text and if you go one way or another you can select more or less text*" - P20). When asked whether they would use *FlexionTouch* on their own mobile device, five participants would reportedly use it, ten would use it if the respective use cases are useful ("*If I can see a benefit in it, then for sure. It really depends on for what it is used*" - P10), and five would rather not use it.

6 DISCUSSION

Inspired by previous tabletop research which presented input methods based on the concept of *imprint-based touch sensing*, we investigated imprint-based input methods for mobile touchscreen devices. Since research efforts on input methods for tabletops cannot be unconditionally applied for smartphones, we conducted two studies in which we first identify potential candidates for imprint-based input methods and a wide range of use cases which support these new inputs. In a second step, we then implemented *FlexionTouch* as a case study to investigate how potential users perform the input, and to gather their first impressions, perceived advantages and disadvantages, and for what they envision to use it.

As a result of the ideation sessions, we identified a wide range of imprint-based input methods that are worth to investigate in future work. Participants envisioned use cases for such input methods that all address shortcomings of current touch input on mobile devices (direct touch with fingertips). Amongst others, this includes providing shortcuts as the recent touch input vocabulary is limited, and techniques to address the fat-finger problem through additional text selection techniques. Despite all the benefits, the interviews also revealed that input methods based on *imprint-based touch sensing* should extend recent touch input instead of replacing it. Direct touch is already intuitive while input techniques based on contact areas are focusing on indirect manipulation since the area is too large for precise target selection (*e.g.*, mapping flexion to a value instead of directly setting it). An evaluation of *FlexionTouch* confirmed this observation. We could not find a statistically significant difference in task completion time between Direct Touch and *FlexionTouch*, but showed that the latter has a low input accuracy. Despite these shortcomings, the post-study interview showed that participants still find *FlexionTouch* helpful and convenient. While the accuracy of *FlexionTouch* is lower than the accuracy of direct touch, our participants were still able to perform the expected input after being trained. Among others, this allows expert users to enter continuous values with just a single touch. This is a first step in bringing imprint-based input techniques to mobile devises e.g smartphones. Our participants envisioned a wide range of scenarios and use cases for which they would use new imprint-based input method for.

In summary, the two presented studies revealed that input methods based on the contact area and shape are useful additions to the limited touch input vocabulary. While we used a commodity smartphone with low-resolution capacitive images that lead to a lower input accuracy (instead of a high-resolution camera-based prototype which does not correspond to a typical mobile device), users stated that the additional input method is helpful and convenient especially for shortcuts and media preview. Future work should explore using different touch sensing technologies (*e.g.*, IR sensing integrated into the LCD similar to the PixelSense technology) to retrieve touch images with a higher resolution. The accuracy and precision of imprint-based input methods can be further improved by applying filters and machine learning techniques. Further, more research is required to understand how to design imprint-based input methods on mobile devices before these can reach the mass-market. With this paper, we contributed the idea and an initial understanding of imprint-based input methods that could naturally extend the touch input vocabulary. We hope that this can be a first step towards extending touch input similar to the promising approaches presented in tabletop research.

7 SOURCE CODE

We provide the implementation of *FlexionTouch* which can be readily deployed on Android devices. These will enable the community to improve and build upon our use case. We hope that the provided prototype can serve as a baseline for further imprint-based input techniques: https://github.com/interactionlab/finger-flexion-interaction.

8 CONCLUSION

In this paper, we investigated input methods based on the contact area and shape of touches which we referred to as *imprint-based touch sensing*. This concept is motivated by successful research in the tabletop domain which presented a wide range of input techniques that leverages the shape of touches. However, this research can not be directly ported to touch-based mobile devices. Therefore, we conducted an ideation session (N=17) to elicit potential imprint-based input methods and their use cases. As a case study, we implemented *FlexionTouch*, an additional input modality that maps the flexion of a flatly placed finger to a continuous value. We evaluated this input method in a subsequent user study (N=20) and found that despite the lower accuracy compared to direct touch (due to capacitive sensing on commodity smartphones), users still find *FlexionTouch* helpful and convenient. We identified that input methods based on *imprint-based touch sensing* are especially beneficial to address current touch input limitations, such as the lack of shortcuts and the fat-finger problem. By releasing the source code of our implementation, we provide the basis for future work to extend our work by investigating the ergonomics of finger flexion on mobile devices, and how the touch input vocabulary can be naturally extended.

REFERENCES

[1] Michelle Annett, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2011. Medusa: A Proximity-aware Multi-touch Tabletop. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. ACM, New York, NY, USA, 337–346. https://doi.org/10.1145/2047196.2047240

[2] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise Selection Techniques for Multi-touch Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal, Quebec, Canada) *(CHI '06)*. ACM, New York, NY, USA, 1263–1272. https://doi.org/10.1145/1124772.1124963

[3] David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. 2013. Extending the Vocabulary of Touch Events with ThumbRock. In *Proceedings of Graphics Interface 2013* (Regina, Sascatchewan, Canada) *(GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 221–228. http://dl.acm.org/citation.cfm?id=2532129.2532166

[4] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* (San Francisco, California, USA) *(MobileHCI '12)*. ACM, New York, NY, USA, 39–48. https://doi.org/10.1145/2371574.2371582

[5] Xiang Cao, A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. 129–136. https://doi.org/10.1109/TABLETOP.2008.4660195

[6] Ashley Colley and Jonna Häkkilä. 2014. Exploring Finger Specific Touch Screen Interaction for Mobile Phone User Interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design* (Sydney, New South Wales, Australia) *(OzCHI '14)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2686612.2686699

[7] Florian Echtler, Manuel Huber, and Gudrun Klinker. 2008. Shadow Tracking on Multi-Touch Tables. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (Napoli, Italy) *(AVI '08)*. Association for Computing Machinery, New York, NY, USA, 388–391. https://doi.org/10.1145/1385569.1385640

[8] Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. 2012. Finger and Hand Detection for Multi-touch Interfaces Based on Maximally Stable Extremal Regions. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces* (Cambridge, Massachusetts, USA) *(ITS '12)*. ACM, New York, NY, USA, 173–182. https://doi.org/10.1145/2396636.2396663

[9] Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2013. ArpèGe: Learning Multitouch Chord Gestures Vocabularies. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (St. Andrews, Scotland, United Kingdom) *(ITS '13)*. ACM, New York, NY, USA, 209–218. https://doi.org/10.1145/2512349.2512795

[10] Hyunjae Gil, DoYoung Lee, Seunggyu Im, and Ian Oakley. 2017. TriTap: Identifying Finger Touches on Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 3879–3890. https://doi.org/10.1145/3025453.3025561

[11] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. ACM, New York, NY, USA, 545–554. https://doi.org/10.1145/2380116.2380184

[12] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. ACM, New York, NY, USA, 59–62. https://doi.org/10.1145/2817721.2817722

[13] Aakar Gupta, Muhammed Anwar, and Ravin Balakrishnan. 2016. Porous Interfaces for Small Screen Multitasking Using Finger Identification. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. ACM, New York, NY, USA, 145–156. https://doi.org/10.1145/2984511.2984557

[14] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 59–70. https://doi.org/10.1145/2858036.2858052

[15] Igor Gvero. 2013. Observing the User Experience, 2Nd Edition: A Practitioner's Guide to User Research by Elizabeth Goodman, Mike Kuniavsky, and Andrea Moed. *SIGSOFT Softw. Eng. Notes* 38, 2 (March 2013), 35–35. https://doi.org/10.1145/2439976.2439993

[16] Jefferson Y. Han. 2005. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) *(UIST '05)*. ACM, New York, NY, USA, 115–118. https://doi.org/10.1145/1095034.1095054

[17] Gunnar Harboe and Elaine M. Huang. 2015. Real-World Affinity Diagramming Practices: Bridging the Paper-Digital Gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 95–104. https://doi.org/10.1145/2702123.2702561

[18] Chris Harrison and Scott Hudson. 2012. Using Shear As a Supplemental Two-dimensional Input Channel for Rich Touchscreen Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. ACM, New York, NY, USA, 3149–3152. https://doi.org/10.1145/2207676.2208730

[19] Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. 2014. TouchTools: Leveraging Familiarity and Skill with Physical Tools to Augment Touch Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 2913–2916. https://doi.org/10.1145/2556288.2557012

[20] Seongkook Heo and Geehyuk Lee. 2011. Force Gestures: Augmented Touch Screen Gestures Using Normal and Tangential Force. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. ACM, New York, NY, USA, 1909–1914. https://doi.org/10.1145/1979742.1979895

[21] Seongkook Heo and Geehyuk Lee. 2011. Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (Stockholm, Sweden) *(MobileHCI '11)*. ACM, New York, NY, USA, 113–122. https://doi.org/10.1145/2037373.2037393

[22] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2869–2881. https://doi.org/10.1145/2858036.2858095

[23] Christian Holz, Senaka Buthpitiya, and Marius Knaust. 2015. Bodyprint: Biometric User Identification on Mobile Devices Using the Capacitive Touchscreen to Scan Body Parts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 3011–3014. https://doi.org/10.1145/2702123.2702518

[24] Amy K. Karlson and Benjamin B. Bederson. 2006. Studies in One-Handed Mobile Design: Habit, Desire and Agility. In *Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4ALL)*.

[25] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. 2012. Interaction Techniques for Unreachable Objects on the Touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (Melbourne, Australia) *(OzCHI '12)*. ACM, New York, NY, USA, 295–298. https://doi.org/10.1145/2414536.2414585

[26] Seiya Koura, Shunsuke Suo, Asako Kimura, Fumihisa Shibata, and Hideyuki Tamura. 2012. Amazing Forearm As an Innovative Interaction Device and Data Storage on Tabletop Display. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces* (Cambridge, Massachusetts, USA) *(ITS '12)*. ACM, New York, NY, USA, 383–386. https://doi.org/10.1145/2396636.2396706

[27] Sven Kratz, Patrick Chiu, and Maribeth Back. 2013. PointPose: Finger Pose Estimation for Touch Input on Mobile Devices Using a Depth Sensor. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (St. Andrews, Scotland, United Kingdom) *(ITS '13)*. ACM, New York, NY, USA, 223–230. https://doi.org/10.1145/2512349.2512824

[28] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3173574.3173934

[29] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2017. A Smartphone Prototype for Touch Interaction on the Whole Device Surface. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI EA '17)*. ACM, New York, NY, USA, Article 100, 8 pages. https://doi.org/10.1145/3098279.3122143

[30] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2018. Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI'18)*. ACM, New York, NY, USA. https://doi.org/10.1145/3173574.3173605

[31] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 779–792. https://doi.org/10.1145/3242587.3242605

[32] Huy Viet Le, Sven Mayer, and Niels Henze. 2019. Investigating the Feasibility of Finger Identification on Capacitive Touchscreens using Deep Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, CA, USA) *(IUI '19)*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3301275.3302295

[33] Yang Li. 2010. Gesture Search: A Tool for Fast Mobile Data Access. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. ACM, New York, NY, USA, 87–96. https://doi.org/10.1145/1866029.1866044

[34] Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental Analysis of Mode Switching Techniques in Pen-Based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 461–470. https://doi.org/10.1145/1054972.1055036

[35] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. 2010. Clutch-Free Panning and Integrated Pan-Zoom Control on Touch-Sensitive Surfaces: The Cyclostar Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2615–2624. https://doi.org/10.1145/1753326.1753724

[36] Nicolai Marquardt, Johannes Kiemer, and Saul Greenberg. 2010. What Caused That Touch?: Expressive Interaction with a Surface Through Fiduciary-tagged Gloves. In *ACM International Conference on Interactive Tabletops and Surfaces* (Saarbrücken, Germany) *(ITS '10)*. ACM, New York, NY, USA, 139–142. https://doi.org/10.1145/1936652.1936680

[37] Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. 2017. WhichFingers: Identifying Fingers on Touch Surfaces and Keyboards Using Vibration Sensors. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. ACM, New York, NY, USA, 41–48. https://doi.org/10.1145/3126594.3126619

[38] Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. 2017. Hand Contact Shape Recognition for Posture-Based Tabletop Widgets and Interaction. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. ACM, New York, NY, USA, 3–11. https://doi.org/10.1145/3132272.3134126

[39] Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. ACM, New York, NY, USA, 220–229. https://doi.org/10.1145/3132272.3134130

[40] Sven Mayer, Michael Mayer, and Niels Henze. 2017. Feasibility Analysis of Detecting the Finger Orientation with Depth Cameras. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. ACM, New York, NY, USA, Article 82, 8 pages. https://doi.org/10.1145/3098279.3122125

[41] Takashi Miyaki and Jun Rekimoto. 2009. GraspZoom: Zooming and Scrolling Control Model for Single-handed Mobile Interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Bonn, Germany) *(MobileHCI '09)*. ACM, New York, NY, USA, Article 11, 4 pages. https://doi.org/10.1145/1613858.1613872

[42] Roderick Murray-Smith. 2017. Stratified, computational interaction via machine learning. In *Eighteenth Yale Workshop on Adaptive and Learning Systems* (New Haven, CT, USA). 95–101.

[43] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. 2012. Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. ACM, New York, NY, USA, 487–496. https://doi.org/10.1145/2380116.2380177

[44] Alexander Ng, John Williamson, and Stephen Brewster. 2015. The Effects of Encumbrance and Mobility on Touch-Based Gesture Interactions for Mobile Phones. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. ACM, New York, NY, USA, 536–546. https://doi.org/10.1145/2785830.2785853

[45] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD. Rasel Islam. 2016. The Flat Finger: Exploring Area Touches on Smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (Santa Clara, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 4238–4249. https://doi.org/10.1145/2858036.2858179

[46] Alex Olwal, Steven Feiner, and Susanna Heyman. 2008. Rubbing and Tapping for Precise and Rapid Selection on Touch-Screen Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 295–304. https://doi.org/10.1145/1357054.1357105

[47] Benjamin Poppinga, Alireza Sahami Shirazi, Niels Henze, Wilko Heuten, and Susanne Boll. 2014. Understanding Shortcut Gestures on Mobile Touch Devices. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services* (Toronto, ON, Canada) *(MobileHCI '14)*. ACM, New York, NY, USA, 173–182. https://doi.org/10.1145/2628363.2628378

[48] A Poston. 2000. Human engineering design data digest. *Washington, DC: Department of Defense Human Factors Engineering Technical Advisory Group* (2000).

[49] Vaughan Pratt. 1987. Direct Least-squares Fitting of Algebraic Surfaces. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. ACM, New York, NY, USA, 145–152. https://doi.org/10.1145/37401.37420

[50] Jun Rekimoto. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI '02)*. ACM, New York, NY, USA, 113–120. https://doi.org/10.1145/503376.503397

[51] Volker Roth and Thea Turner. 2009. Bezel Swipe: Conflict-free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 1523–1526. https://doi.org/10.1145/1518701.1518933

[52] Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. MicroRolls: Expanding Touch-screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 927–936. https://doi.org/10.1145/1518701.1518843

[53] Ben Schneiderman. 1986. Eight golden rules of interface design. *Disponible en* (1986).

[54] Robin Schweigert, Jan Leusmann, Simon Hagenmayer, Maximilian Weiundefined, Huy Viet Le, Sven Mayer, and Andreas Bulling. 2019. KnuckleTouch: Enabling Knuckle Gestures on Capacitive Touchscreens Using Deep Learning. In *Proceedings of Mensch Und Computer 2019* (Hamburg, Germany) *(MuC'19)*. ACM, New York, NY, USA, 387–397. https://doi.org/10.1145/3340764.3340767

[55] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2017. Experimental Analysis of Mode Switching Techniques in Touch-Based User Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3267–3280. https://doi.org/10.1145/3025453.3025865

[56] Jingtao Wang and John Canny. 2004. FingerSense: Augmenting Expressiveness to Physical Pushing Button by Fingertip Identification. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria) *(CHI EA '04)*. ACM, New York, NY, USA, 1267–1270. https://doi.org/10.1145/985921.986040

[57] Andrew D. Wilson. 2010. Using a Depth Camera As a Touch Sensor. In *ACM International Conference on Interactive Tabletops and Surfaces* (Saarbrücken, Germany) *(ITS '10)*. ACM, New York, NY, USA, 69–72. https://doi.org/10.1145/1936652.1936665

[58] Raphael Wimmer and Sebastian Boring. 2009. HandSense: Discriminating Different Ways of Grasping and Holding a Tangible User Interface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (Cambridge, United Kingdom) *(TEI '09)*. Association for Computing Machinery, New York, NY, USA, 359–362. https://doi.org/10.1145/1517664.1517736

[59] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (Madeira, Portugal) *(ITS '15)*. ACM, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737

[60] T. Y. Zhang and C. Y. Suen. 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. *Commun. ACM* 27, 3 (March 1984), 236–239. https://doi.org/10.1145/357994.358023

[61] Jingjie Zheng and Daniel Vogel. 2016. Finger-Aware Shortcuts. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 4274–4285. https://doi.org/10.1145/2858036.2858355